

DOCUMENT RESUME

ED 161 418

IR 006 258

AUTHOR Franks, Edward W.
TITLE CBES--An Efficient Implementation of the Coursewriter Language.
PUB DATE Mar 78
NOTE 9p.; Paper presented at the Annual Meeting of the Association for the Development of Computer Based Instructional Systems (Dallas, Texas, March 1-4, 1978); For related document, see IR 006 231

EDRS PRICE MF-\$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS *Computer Assisted Instruction; *Computer Programs; Flow Charts; Instructional Design; *Instructional Systems; *Programming Languages; Systems Analysis; *Systems Development

IDENTIFIERS *Coursewriter III; Ohio State University

ABSTRACT

An extensive computer based education system (CBES) built around the IBM Coursewriter III program product at Ohio State University is described. In this system, numerous extensions have been added to the Coursewriter III language to provide capabilities needed to implement sophisticated instructional strategies. CBES design goals include lower CPU usage per student interaction, fewer disk accessions per student interaction, all existing course materials to operate without alteration, system modifications to be transparent at the student level, standard Coursewriter III authoring capabilities to be maintained, and provision for enhancements to be made where practical. Other sections of the paper are concerned with system operation logic, distribution of processing, redundant processing, reduction of disk activity, locating and retrieving information, implementation, and enhancements. (VT)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

ED161418

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

CBES -- AN EFFICIENT IMPLEMENTATION
OF THE COURSEWRITER LANGUAGE

by

Edward W. Franks

"PERMISSION TO REPRODUCE THIS
MATERIAL IN MICROFICHE ONLY
HAS BEEN GRANTED BY

John J. Hirschbuhl

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC) AND
USERS OF THE ERIC SYSTEM."

IR006258

**CBES -- An Efficient Implementation
of the Coursewriter Language**

**Edward W. Franks
The Ohio State University**

During the past eight years The Ohio State University has developed an extensive computer based education system built around the IBM Coursewriter III program product. Numerous extensions have been added to the Coursewriter III language to provide capabilities needed to implement sophisticated instructional strategies. Currently there are fifty operational courses being used by both undergraduate and graduate level university students. These courses may be accessed by a maximum of eighty simultaneous users with as many as 70,000 student interactions provided by the system in a sixteen hour operating day.

The CBE activity takes place on an IBM 370 model 158 computer system. In addition to CBE, this computer provides concurrent processing for three other on-line systems and a variety of batch applications required for routine university operation.

During the Spring Quarter of 1976 CBE activity was consuming over 70% of the CPU at peak usage times with resultant degradation of service to both CBE and non-CBE users alike. The average CBE student interaction was calculated to take .4 of a second CPU time. This high CPU usage, coupled with a disk accession rate of over two million per day, led to a growing concern among those responsible for maintaining computer services. As computer operations essential to university operation were increasingly degraded, it became apparent that action of an immediate and drastic nature would be required.

Initial stop-gap measures included a curtailment of CBE development and the diverting of student usage toward non-peak hours. As a result of these measures, a state of equilibrium was achieved; a poor but acceptable system response time for all university on-line systems could be maintained during the entire working day.

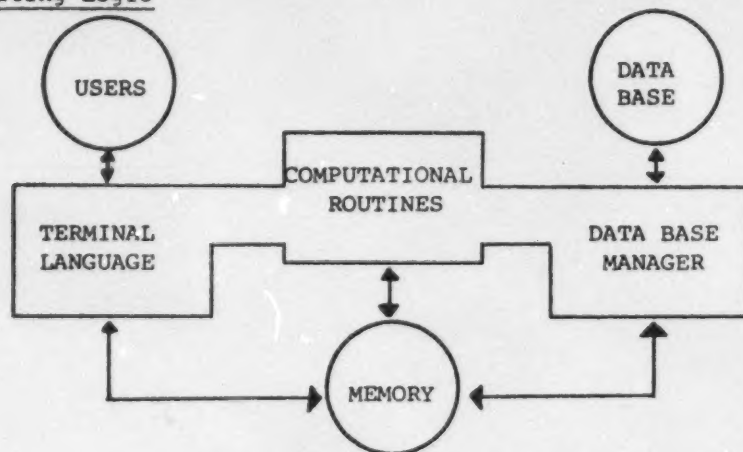
A general CBE retrenchment was considered, whereby some of the educationally less valuable courses would be removed from the system. This approach was abandoned in the realization that these courses were being only lightly used and that their removal would have little effect upon the problem. The acquisition of a faster computer was also discussed, but lack of available funding forced this approach to be abandoned also.

With the opinion that .4 of a second CPU per student interaction -- roughly 300,000 machine instructions -- was excessive, a study was begun of IBM's Coursewriter III program product to determine in what areas it could be made more efficient. A few week's study of the program, however, indicated that to achieve the efficiency gains necessary for the continued survival and growth of The Ohio State University CBE program, nothing short of a complete redesign and recoding of the system program would be required. Such action, it was hoped, would result in a significant reduction in computer resources used by the CBE system. In July of 1976 work was begun on the Computer Based Education System (CBES).

CBES Design Goals

1. lower CPU usage per student interaction
2. fewer disk accessions per student interaction
3. all existing course materials to operate without alteration
4. system modifications to be transparent at the student level
5. standard Coursewriter III authoring capabilities to be maintained
6. provision for enhancements to be made where practical

System Operating Logic



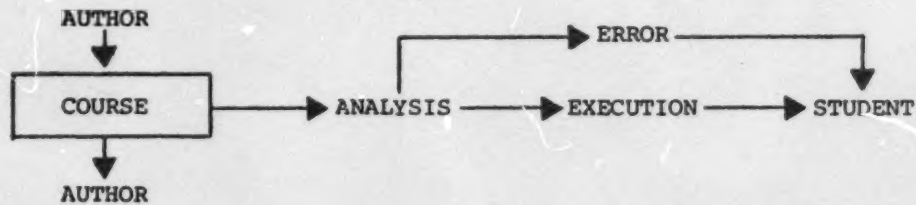
A CBE system can be divided into three physical groups: users, a data base, and memory. While processing a student interaction, information passes between the user and memory and between the data base and memory. While in the memory the information can be processed by the computational

routines. As each student interaction begins, a portion of the memory is assigned as a work area and information is passed from the user. The computational routines then process the information, formulating a response to be returned to the user. During the computational phase information will be transferred as necessary between the data base and memory. Upon completion of the computational phase, a response will be returned to the user and the memory required for work space released to other users. Since the information transfers occur asynchronously, several can overlap; while one user is awaiting completion of an information transfer, another can be actively processing. Due to the sequential execution of the Coursewriter language, each user need have only one transfer in progress at a given time. As each transfer completes, the computational routine must scan the list of users until it finds the one ready to continue processing. A system designed for 100 users will average 50 checking operations for each information transfer. Assuming a minimum of 2 user transfers and 10 data base transfers per student interaction, $(2 + 10) \times 50 = 600$ checking operations would be required for each student interaction. Since a system designed for 200 users would require $(2 + 10) \times 100 = 1200$ checking operations per student interaction, the total check activity increases exponentially with the addition of users. The only time information transfer from the data base occurs, however, is during the time the user occupies a memory work space. Assuming 10 such work spaces, it is possible to associate data base information transfers with a work space, rather than with a specific user. Utilizing this strategy, 100 checking operations will be required for the 2 user information transfers but only 50 checking operations for the data base information transfers for a total of $(2 \times 50) + (10 \times 5) = 150$ checking operations for each student interaction.

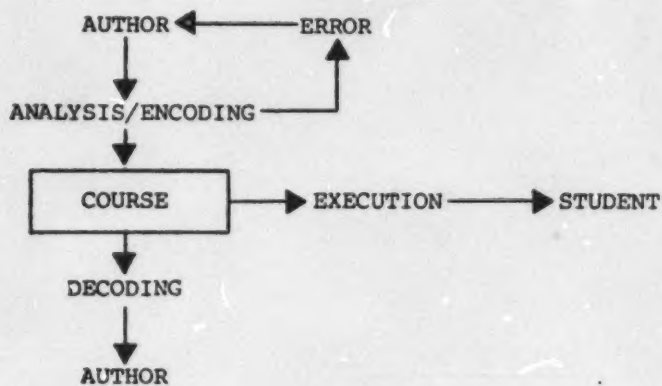
Distribution of Processing

A course written in the Coursewriter language consists of a series of instructions stored in the data base. The computation section fetches each instruction, in turn, and performs the action appropriate to it. These statements originate at the author's terminal. The author in the IBM Coursewriter III system enters the statement, the system performs syntax checking and the statement is stored in the data base. This statement can then be retrieved by the author for later reference. During student execution the computational routine fetches the statement, analyzes it, checks for errors, and if no errors are found, executes it. In CBES, the statement analysis is shifted to the author interface. Consequently, most errors are isolated for immediate correction by the author. In addition, statement analysis need be performed only once, rather than each time the statement is executed.

Coursewriter III



Computer Based Education System



Redundant Processing

A major method of reducing the computer resources required to support CBE is in minimizing the calculation of often-required values. One such value is the current time of day. It is not unusual for the time, calculated to the nearest minute, to be required for each student interaction. By arranging for the system to calculate the current time only once each minute and saving the result, several hundred calculations are saved each minute.

Reduction of Disk Activity

There are two reasons to fetch information from a data base: (1) because it is needed, and (2) because a copy previously fetched has been overlaid. The first is under the control of the author. The

second, while it cannot be eliminated, can be reduced at the system level. Under CBES memory work space can accommodate 4 blocks of course material. When a new block is needed, it overlays the oldest block, allowing retention of the newer ones. An exception to this is a block that will not be reused. These blocks are always overlayed first, allowing retention of 3 potentially reuseable blocks. A fifth block is also provided for non-course material fetches. An additional provision has been made to permanently retain frequently used course blocks in a non-user memory area where all users may access a single copy. This is particularly useful with courses that will have 20 or more simultaneous users.

Locating and Retrieving Information

In a large data base, a major consideration is the ability to find a particular information block quickly and efficiently. To aid in the search, directories are maintained by the system. As the directories become larger, more time is required to search them. To minimize directory search time a technique known as "hashing" is employed. In hashing, the item to be searched is reduced to a numeric value and then divided by the total "slots" in the table. The remainder is the slot number in which to begin the search.

Implementation

Using the aforementioned techniques and principles, CBES was designed and programmed over an 18 month period culminating with its implementation during the Autumn Quarter, 1977. Approximately one-half of the CBE load was assigned to CBES. Operation was very good throughout the quarter, with response times remaining nearly instantaneous even under peak usage. Both down time and instabilities were conspicuous in their absence. Although performance statistics are incomplete, initial studies indicate a reduction in both CPU and disk activity of well over 80 percent.

Enhancements

In addition to reducing the amount of computer resources required, it was desired to increase the accessibility of CBES to both the author and supervisor. Following is a list of enhancements found in CBES:

1. elimination of monitor mode
All Coursewriter monitor functions are available in author mode.
2. author capabilities available to supervisors
Supervisor-as-author mode simplifies system maintenance.
3. multiple authors
A system of interlocks allows the simultaneous sign on of multiple authors on each course.

4. terminal identifier support
Four character terminal IDs are provided for accurate terminal usage statistics.
5. password protection
Each student record can be optionally protected with an 8-character password.
6. author-as-student
The author can specify the use of any student record. By specifying "restart" the author emulates a student sign on.
7. load/show commands expanded
All areas of the student record are now accessible.
8. author "debug" mode
An author can "step" through his program, inspecting and altering his storage facilities at will.
9. control of student sign off
A student entering "sign off" is branched to label 'end' where the author can provide sign off processing. An additional parameter to inhibit student sign off has also been provided.
10. variable length buffer type
Position and length may be specified when a buffer is the object of a "ty" operation.
11. additional editing parameters provided
12. editing and prompting during student sign on
13. accountability
A record of resources required by each user is maintained.
14. stability
Extensive error recovery and loop detection routines assure system stability.

The development of CBES was forced by the unavailability of other alternatives. At the outset there existed considerable doubt that such a project could be completed with the necessary system stability and savings of resources. As the conversion nears completion, however, it has become apparent that all design goals have been met or exceeded and that the cost-effectiveness of CBE at The Ohio State University has been greatly increased.

Acknowledgements

In a project of this size and duration a number of people, by necessity, become involved. Considerable programming was done by Beverly

B. Roark, University Systems. Debugging of the finished modules was greatly assisted by Thomas B. Lloyd, CBE design consultant. In addition, the author would like to extend his personal thanks to the following for their counsel, advice, and support:

Carl M. Philabaum, design consultant, Computer Based Education
Salvatore M. Abate, operations director, Computer Based Education
Keith A. Hall, director, Computer Based Education
James B. Randels, director of systems programming, University
Systems
J. Carroll Notestine, director, University Systems
Glynis L. Getman, secretary, Computer Based Education